Vision, Modeling, and Visualization (2018)
F. Beck, C. Dachsbacher, and F. Sadlo (Eds.)

Supplement to Paper: Fast and Dynamic Construction of Bounding Volume Hierarchies based on Loose Octrees

Feng Gu Johannes Jendersie Thorsten Grosch

TU Clausthal, Germany

1. The Calculation of the DFS Key

The depth first search (DFS) ordering key for a node with level l in the normal binary tree and Morton code m can be calculated as

$$k = k_1 + k_2 + k_3$$

where k_1 is the number of visited nodes with levels less than l, k_2 is the number of visited nodes with levels equal l, k_3 is the number of visited nodes with levels larger than l by the DFS of the normal binary tree before visiting the current node. It is easy to see

$$k_1 = l + \sum_{i=1}^{l} \left\lfloor \frac{m}{2^i} \right\rfloor = l + m - \mathrm{popc}(m) \tag{1}$$

and

$$k_2 = m \tag{2}$$

and

$$k_3 = m \sum_{i=1}^{L'-l} 2^i = m(2^{L'-l+1} - 2) \tag{3}$$

From equations (1), (2) and (3) we have

$$\begin{aligned} k &= l + m - \mathrm{popc}(m) + m + m(2^{L'-l+1} - 2) \\ &= l - \mathrm{popc}(m) + m2^{L'-l+1} \\ &= l - \mathrm{popc}(m') + 2m' \end{aligned} \tag{4}$$

The multiplication in the second last line is equal to a left shift of $L' - l + 1$ binary digits which is exactly our definition of the adjusted Morton code ($m' = m \ll (L' - l)$) times two. ■

2. Theorem: $\theta(i, j)$ can be used instead of $\eta(i, j)$

Proof: In the algorithm there are only comparisons of $\eta(i, j)$ with $\eta(i,k)$ that have a common term i with j and k coming from different sides of i. Note that $\eta(i, j) = \min\{l_i, \theta(i, j)\}$ and $\eta(i,k) = \min\{l_i, \theta(i,k)\}$. Therefore, $\theta(i, j) = \theta(i,k) \Rightarrow \eta(i, j) = \eta(i,k)$ holds trivially.

If $\theta(i, j) > \theta(i,k)$, there is either $\eta(i, j) > \eta(i,k)$ or $\eta(i, j) = \eta(i,k)$. The latter case $\eta(i, j) = \eta(i,k)$, implies that $l_i \leq \theta(i,k)$, i.e. j,k are descendants of i and are therefore both on the same side of i which cannot happen. Thus, it is safe to replace η with θ in the algorithm. ■

3. Theorem: $\theta(i,i-1) = \theta(i, j)$ with $j > i$ only happens when i, j are descendants of $i-1$ but j is not the descendant of i

Proof: Let p be the common prefix for $i-1, i$ and j with length $\theta(i,i-1) = \theta(i, j)$. First, if $\theta(i,i-1) = \theta(i, j)$ with $j > i$, then $i-1$ must be the ancestor of i. If this is not the case, then $\theta(i,i-1) < l_{i-1}$, with $i-1$ having the prefix $p0$ and i the prefix $p1$. However, $\theta(i,i-1) = \theta(i, j)$ implies either $l_j = \theta(i, j)$ or j has prefix $p0$, but $l_j = \theta(i, j)$ implies that an ancestor is after its descendants and prefix $p0$ implies $m'_j < m'_i$ which are both not possible.

After knowing that $i-1$ is the ancestor of i, we have $\theta(i,i-1) = l_{i-1}$. Combining with $\theta(i,i-1) = \theta(i, j)$ we know j is also a descendant of $i-1$.

Finally, we need to prove i is not an ancestor of j: if this would be the case, then $\theta(i, j) = l_i$. However, since i is a descendant of $i-1$, we have $l_i > l_{i-1}$. This implies $\theta(i, j) = l_i > l_{i-1} = \theta(i,i-1)$. ■